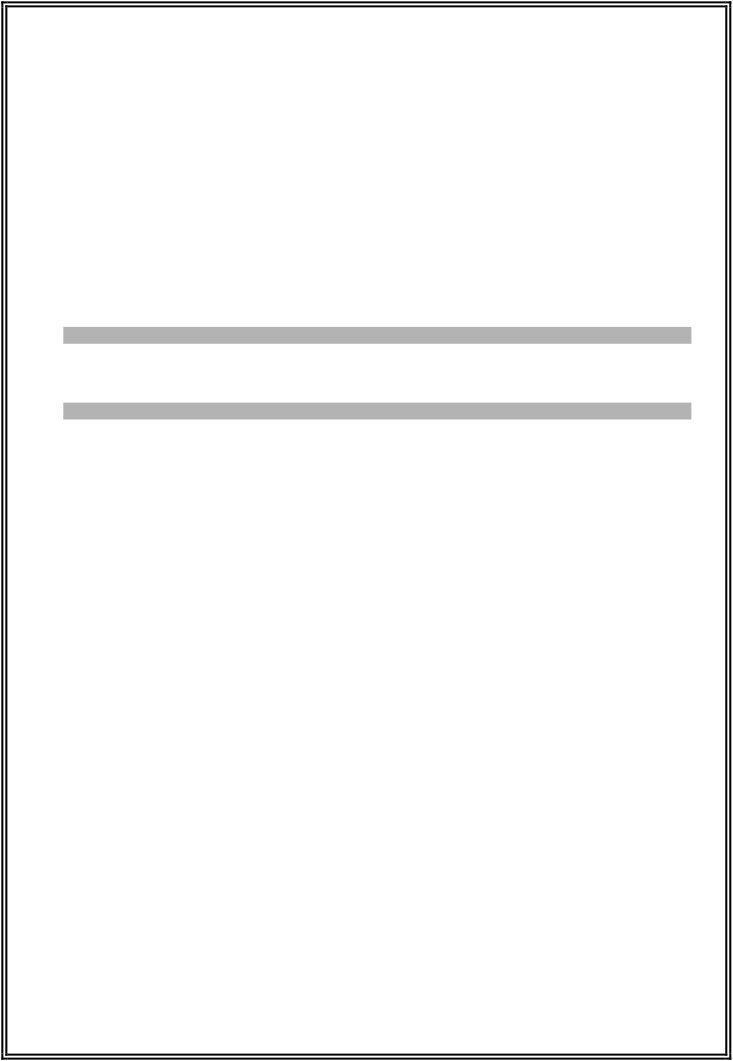
**Evaluation Only. Created with Aspose.Words. Copyright 2003-2023 Aspose Pty Ltd.**



# **GCE A LEVEL MARKING SCHEME**



**SUMMER 2019**

**A LEVEL (NEW)**

**COMPUTER SCIENCE - UNIT 3 1500U30-1**

# **INTRODUCTION**

This marking scheme was used by WJEC for the 2019 examination. It was finalised after detailed discussion at examiners' conferences by all the examiners involved in the assessment. The conference was held shortly after the paper was taken so that reference could be made to the full range of candidates' responses, with photocopied scripts forming the basis of discussion. The aim of the conference was to ensure that the marking scheme was interpreted and applied in the same way by all examiners.

It is hoped that this information will be of assistance to centres but it is recognised at the same time that, without the benefit of participation in the examiners' conference, teachers may have different views on certain matters of detail or interpretation.

WJEC regrets that it cannot enter into any discussion or correspondence about this marking scheme.

**GCE A LEVEL (NEW) COMPUTER SCIENCE - UNIT 3**

**SUMMER 2019 MARK SCHEME**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question** | **Answer** | | | | | | | | | | | | | | | | | | | | | | | | **Mark** | | **AO1** | | **AO2** | | **AO3** | **Total** |
| 1(a) | **One mark for each principle plus one for the description**   * A stack uses the last in first out (LIFO) principle. * In a stack the last or most recent item of data to be added to the stack is removed first. * Adding data to a stack is known as pushing, whilst removing data from a stack is known as popping. * A queue uses the first in first out (FIFO) principle. In a queue the last or most recent item of data added to a queue is the last to be removed.   Give credit for references to pointers. | | | | | | | | | | | | | | | | | | | | | | | | 1 1  1  1 1 | | 1.b | |  | |  | 4 |
| 1(b)(i) | A C G L  E  Award 1 mark for node in correct position  Award 1 mark for both arrows to and from correct nodes. | | | | | | | | | | | | | | | | | | | | | | | | 1 1 | |  | | 2.1 | |  | 2 |
| 1(b)(ii) | A | | |  | | | | | E | | |  | | | G | | |  | | | L | | | | 1 1 | |  | | 2.1 | |  | 2 |
|  | | |  | | |  | | |  | | | |
|  | |  | |  | | |  | | |  | | |  | | |  | | |  | | |  | |  | |
|  |  |  | |  | |  | | |  | | |  | | |  | | |  | | |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Question** | **Answer** | **Mark** | **AO1** | **AO2** | **AO3** | **Total** |
| 2(a) | Correct answer can be established using different steps / laws / rules / identities / dual relations.  One possible solution: A.A + A.B + A.B + B.B A.B + A.B + B.B  A.B + A.B  A.(B + B)  A.1  A  Correctly applying identities to arrive at correct answer 5 marks  Correctly applying identities but arriving at wrong answer 1 mark for each correct step up to a maximum of 4 | 5 |  | 2.1 |  | 5 |
| 2(b) | Correct answer can be established using different steps / laws / rules / identities / dual relations.  One possible solution: A.B + A.C + B  A + B + A.C + B  A + A.C + B + B  A + A.C  1 + C  1  Correctly applying identities to arrive at correct answer 5 marks  Correctly applying identities but arriving at wrong answer 1 mark for each correct step up to a maximum of 4 | 5 |  | 2.1 |  | 5 |
| 3 | 01101001  2  10000000  2  AND 00000000  State of left hand bit is 0  Award 1 mark for using an AND or XOR mask. Award 1 mark for correct result  Award 1 mark for determining the state of the most significant left hand bit. | 1 1 1 |  | 2.1 |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Question** | **Answer** | **Mark** | **AO1** | **AO2** | **AO3** | **Total** |
| 4(a) | A natural language interface is where speech and linguistics is used (1 mark) to interact and control a software application. (1 mark) | 2 | 1.b |  |  | 2 |
| 4(b) | **Indicative content**  One potential use for a natural language interface would be in translation software. Natural language could be processed in real time to allow for a seamless translation service.  1 mark for suitable example 1 mark for justification | 1 1 | 1.b |  |  | 2 |
| 4(c) | **One mark per example up to a maximum of 3 marks.**  Colloquialisms and words can be interpreted differently regionally.  Accents could make is difficult for a natural language interface to identify the words being spoken.  Ambiguity in spoken language where a word may have more than one interpretation.  Background noise could cause problems. Illness such as sore throat | 1 1 1  1 1 | 1.b |  |  | 3 |
| 5(a) | The algorithm sorts an array (1 mark) into ascending order (1 mark). | 2 |  | 2.1 |  | 2 |
| 5(b) | This a recursive (1 mark) algorithm that calls itself (1 mark) and has a stopping condition (1 mark). | 3 |  | 2.1 |  | 3 |
| 5(c) |

**This document was truncated here because it was created in the Evaluation Mode.**